

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2001-325248

(43)Date of publication of application : 22.11.2001

(51)Int.Cl.

G06F 17/21

(21)Application number : 2000-144947 (71)Applicant : FUJI XEROX CO LTD

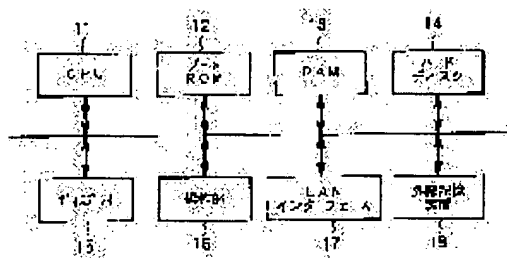
(22)Date of filing : 17.05.2000 (72)Inventor : IWATA NOBUO

(54) DOCUMENT DATA PROCESSOR

(57)Abstract:

PROBLEM TO BE SOLVED: To provide a document data processor having improved processing efficiency in contrast to a conventional document data processor which has a problem that the processing efficiency is low since the processing of an HTML parser is performed after the processing of an XML parser is performed.

SOLUTION: In this document data processor, a CPU 11 reads document data and performs the processing as the XML parser. At the time of detecting a tag which is not an XML tag during the processing of the XML parser, the start tag processing part or end tag processing part of the HTML parser is activated and the pertinent part is processed. Further, at the time of finding the tag related to CDATA or a pre-format, a corresponding processing is performed.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of

BEST AVAILABLE COPY

rejection}.

[Kind of final disposal of application other
than the examiner's decision of rejection or
application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's
decision of rejection]

[Date of requesting appeal against
examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2001-325248

(P2001-325248A)

(43) 公開日 平成13年11月22日 (2001. 11. 22)

(51) Int. Cl.⁷

G 0 6 F 17/21

識別記号

5 0 1

F I

G 0 6 F 17/21

フォーマット (参考)

5 0 1 T 5 B 0 0 9

審査請求 未請求 請求項の数 9 O L (全 10 頁)

(21) 出願番号 特願2000-144947 (P2000-144947)

(22) 出願日 平成12年 5 月17日 (2000. 5. 17)

(71) 出願人 000005496

富士ゼロックス株式会社

東京都港区赤坂二丁目17番22号

(72) 発明者 岩田 伸夫

神奈川県海老名市本郷2274番地 富士ゼロ

ックス株式会社海老名事業所内

(74) 代理人 100075258

弁理士 吉田 研二 (外 2 名)

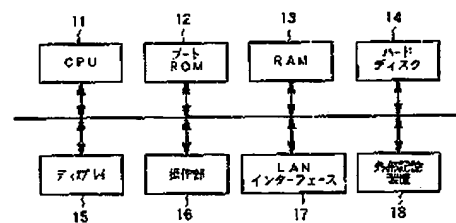
Fターム (参考) 5B009 NA05

(54) 【発明の名称】 文書データ処理装置

(57) 【要約】

【課題】 従来の文書データ処理装置では、XMLパーザの処理が行われた後にHTMLパーザの処理が行われるので、処理効率が低いという問題点があったが、本発明では、処理効率を向上できる文書データ処理装置を提供する。

【解決手段】 CPU 11 が文書データを読み込んで、XMLパーザとしての処理を行い、このXMLパーザの処理中に、XMLタグでないタグを検出すると、HTMLパーザの開始タグ処理部又は終了タグ処理部を起動して当該部分を処理し、さらにC D A T A又はプレフォーマットに関連するタグであるときには、対応する処理を行う文書データ処理装置である。



【特許請求の範囲】

【請求項1】 HTMLにより記述された部分文書データとXMLにより記述された部分文書データの少なくとも一方を含むタグ付き文書データ、を処理する文書データ処理装置であって、

所定の部分文書データをHTML文書として処理し、出力するHTMLパーザ手段と、
処理対象となったタグ付き文書データを部分文書データごとに順次解析し、解析の結果に応じて前記HTMLパーザ手段を起動する動作と、当該解析された部分文書データをXML文書として処理する動作とのいずれかを選択的に行うXMLパーザ手段と、

を含むことを特徴とする文書データ処理装置。

【請求項2】 請求項1に記載の文書データ処理装置であって、さらに、
処理対象となったタグ付き文書データのデータ型属性を取得する手段と、

前記データ型属性に対応付けて事前に定義されているデフォルトデータ型定義を取得する手段とを含み、

前記XMLパーザ手段が、前記取得したデフォルトデータ型定義に従って処理を行うことを特徴とする文書データ処理装置。

【請求項3】 請求項2に記載の文書データ処理装置であって、

前記XMLパーザ手段は、処理対象となったタグ付き文書データにXML宣言がない場合にのみ前記取得したデフォルトデータ型定義に従って処理を行うことを特徴とする文書データ処理装置。

【請求項4】 請求項1に記載の文書データ処理装置であって、

処理対象となったタグ付き文書データと、当該文書データに関連づけられたデータ型定義とを取得する手段と、
前記タグ付き文書データのデータ型属性を取得する手段と、

前記データ型属性に対応付けて事前に定義されているデフォルトデータ型定義を取得する手段とを含み、

前記XMLパーザ手段が、前記取得したデータ型定義とデフォルトデータ型定義とに従って処理を行うことを特徴とする文書データ処理装置。

【請求項5】 請求項1から4のいずれかに記載の文書データ処理装置において、

特殊タグごとに、当該特殊タグに関連する部分文書データを処理するパーザ手段と、

前記特殊タグの情報と、当該特殊タグに対応するパーザ手段とを少なくとも一組設定する手段を含み、

前記XMLパーザ手段は、タグ付き文書データを解析し、当該解析の結果、前記特殊タグを検出すると、当該特殊タグに対応するパーザ手段を起動することを特徴とする文書データ処理装置。

【請求項6】 請求項5に記載の文書データ処理装置に

おいて、

前記特殊タグには、少なくともCDATAに関連するタグと、プレフォーマットに関連するタグとのいずれかを含むことを特徴とする文書データ処理装置。

【請求項7】 請求項1から6のいずれかに記載の文書データ処理装置において、

さらに、各タグごとの省略可否情報を格納する手段を含み、

前記XMLパーザ手段は、省略可能に設定されたタグを検出すると、タグが省略されているか否かを解析し、当該解析の結果に基づいてタグが省略されているときには、当該省略されたタグを補充して文書データを処理することを特徴とする文書データ処理装置。

【請求項8】 第1のルールで記述された部分文書データと第2のルールに従って記述された部分文書データの少なくとも一方を含む文書データ、を処理する文書データ処理装置であって、
所定の部分文書データを前記第1のルールに従って処理し、出力する第1パーザ手段と、

処理対象となった文書データを部分文書データごとに順次解析し、解析の結果に応じて前記第1パーザ手段を起動する動作と、当該解析された部分文書データを第2のルールに従って処理する動作とのいずれかを選択的に行う第2パーザ手段と、

を含むことを特徴とする文書データ処理装置。

【請求項9】 第1のルールで記述された部分文書データと第2のルールに従って記述された部分文書データの少なくとも一方を含む文書データ、を処理する文書データ処理プログラムであって、

所定の部分文書データを前記第1のルールに従って処理し、出力する第1パーザモジュールと、

処理対象となった文書データを部分文書データごとに順次解析し、解析の結果に応じて前記第1パーザモジュールを起動する動作と、当該解析された部分文書データを第2のルールに従って処理する動作とのいずれかを選択的に行う第2パーザモジュールと、

を含む文書データ処理プログラムを格納したことを特徴とするコンピュータ読み取り可能な記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、構造化された文書データを表示等するための文書データ処理装置に係り、特にXML (Extensible Markup Language) と、HTML (HyperText Markup Language) との双方に対応可能な文書データ処理装置に関する。

【0002】

【従来の技術】近年、インターネットで広く用いられる文書データの形式としてHTMLと呼ばれるマーク付け言語がある。HTMLは、SGML (Standard Generalized Markup Language) と呼ばれる、メタ言語文法に準

鍵して記述されている。しかし、このSGMLにより記述された文書の処理は、種々の指定処理が可能であるものの、処理が複雑になるため、XMLと呼ばれる簡略化された言語が策定されつつある。また、このXMLには、文書中に文書の属性や特定の処理の対象となるデータ（直接表示はされないが処理装置で利用されるデータ）を含ませることができる。

【0003】このXMLにより記述された文書データは、HTMLと同様に取得されて処理されるのであるが、従来の文書データ処理装置としてのWebブラウザを含むパーソナルコンピュータでは、このXMLを直接処理できないため、取得したXML文書を、HTMLに変換する処理を行ってから、HTML文書として改めて処理を行っている。

【0004】ところで、XMLやSGMLでは、文書ファイル中または文書ファイルから参照される他のファイルにDTD（Data Type Definition）を設定してシンタックスを宣言し、この宣言に従って文書データの処理を行わせる。

【0005】例えば、SGMLでは、マーク付けのためのタグ（特定の処理を行わせるべきコンテンツを囲むデータ）をDTDを用いて定義できる。このSGMLでは、終了を表すタグを省略して記載できるようにDTDを用いた定義を作成できる。このように終了のタグを省略した文書データを処理する場合、当該タグに対するコンテンツであるか否かを判別し、そうでないコンテンツが見いだされた場合に、終了タグがあったものとして処理する必要がある。具体的にHTMLの<P>（段落の開始タグ）は、対応する終了タグ（</P>）を省略できることをDTDを用いて定義できる。さらにSGMLでは、文書データ中に必須の開始タグを省略可能と定義することもできるようになっている。

【0006】これに比べ、XMLは、開始タグと終了タグの種類が制限されるとともに、コンテンツのタイプも参照データとタグとの関係でSGMLより限定的に定められている。すなわち、SGMLでは、コンテンツ中のタグや（画像データなどに対する）参照データを処理するPCDATA（Parsed Character Data）と、タグを無視して参照データのみを処理するRCDATA（Rapidly Accessible Character Data）と、タグも参照データも単なる文字列として扱うことができるCDATA（Character Data）との3種類があったが、XMLでは、PCDATAのみが利用可能である。ここで、CDATAは、表示や印刷の目的でなく、JavaScriptによるスクリプトプログラムの記述に利用されている。

【0007】このため、従来の文書データ処理装置では、XMLとHTMLの双方に対応させるために、XMLデータを処理してHTMLデータに変換し、次に変換後のHTMLデータを処理することとしている。

【0008】

【発明が解決しようとする課題】このように、上記従来の文書データ処理装置では、XMLをHTMLへ変換するためにXMLパーザを実行し、さらにこのパーザの処理結果に基づいてHTMLパーザを実行するので、処理負荷が大きくなる。また、既存のHTML文書の多くは、DTD定義が関連づけられていないものが多く、XMLパーザが正常に動作しない場合がある。さらに、既存のHTML文書では、終了タグが省略されているものが多く、XMLパーザがそのまま正常に処理できないという問題点があった。

【0009】さらに、近年のHTML文書データには、JavaScriptのための<SCRIPT>タグや、スタイルシートの指定のための<STYLE>タグ等が設定されており、コンテンツがCDATAとして処理されることが前提となっているものがある。さらに、<PRE>タグと呼ばれるタグを用いて、スペースや改行をコンテンツ通りに表示する手法が用いられたものもある。これらの文書データは、XMLパーザではそのまま正常に処理することができない。

【0010】本発明は上記実情に鑑みて為されたもので、HTML中に混在するXMLデータを正常に処理でき、かつ処理効率を向上できる文書データ処理装置を提供することを目的とする。

【0011】

【課題を解決するための手段】上記従来例の問題点を解決するための請求項1記載の発明は、文書データ処理装置であって、HTMLにより記述された部分文書データとXMLにより記述された部分文書データの少なくとも一方を含むタグ付き文書データ、を処理する文書データ処理装置であって、所定の部分文書データをHTML文書として処理し、出力するHTMLパーザ手段と、処理対象となったタグ付き文書データを部分文書データごとに順次解析し、解析の結果に応じて前記HTMLパーザ手段を起動する動作と、当該解析された部分文書データをXML文書として処理する動作とのいずれかを選択的に行うXMLパーザ手段と、を含むことを特徴としている。

【0012】このXMLパーザ手段により、XMLパーザの解析処理中に検出されたHTML部分の文書データがHTMLパーザにより処理され、処理効率を向上できる。また、上記従来例の問題点を解決するための請求項2記載の発明は、請求項1記載の文書データ処理装置において、処理対象となったタグ付き文書データのデータ型属性を取得する手段と、前記データ型属性に対応付けて事前に定義されているデフォルトデータ型定義を取得する手段とを含み、前記XMLパーザ手段が、前記取得したデフォルトデータ型定義に従って処理を行うことを特徴としている。これにより、DTD定義が関連づけられていない文書データに対してもデフォルトのDTDを適用でき、正常に文書データを処理できる。

【0013】さらに、XMLパーザ手段は、処理対象となったタグ付き文書データにXML宣言がない場合にのみ前記取得したデフォルトデータ型定義に従って処理を行うことも好適である。

【0014】また、上記従来例の問題点を解決するための請求項4記載の発明は、請求項1に記載の文書データ処理装置であって、処理対象となったタグ付き文書データと、当該文書データに関連づけられたデータ型定義とを取得する手段と、前記タグ付き文書データのデータ型属性を取得する手段と、前記データ型属性に対応付けて事前に定義されているデフォルトデータ型定義を取得する手段とを含み、前記XMLパーザ手段が、前記取得したデータ型定義とデフォルトデータ型定義とに従って処理を行うことを特徴としている。

【0015】さらに、上記従来例の問題点を解決するための請求項5記載の発明は、請求項1から4のいずれかに記載の文書データ処理装置において、特殊タグごとに、当該特殊タグに関連する部分文書データを処理するパーザ手段と、前記特殊タグの情報と、当該特殊タグに対応するパーザ手段とを少なくとも一組設定する手段を含み、前記XMLパーザ手段は、タグ付き文書データを解析し、当該解析の結果、前記特殊タグを検出すると、当該特殊タグに対応するパーザ手段を起動することを特徴としている。

【0016】また、ここで前記特殊タグには、少なくともCDATAに関連するタグと、プレフォーマットに関連するタグとのいずれかを含むことが好ましい。

【0017】さらに、上記従来例の問題点を解決するための請求項7記載の発明は、請求項1から6のいずれかに記載の文書データ処理装置において、さらに、各タグごとの省略可否情報を格納する手段を含み、前記XMLパーザ手段は、省略可能に設定されたタグを検出すると、タグが省略されているか否かを解析し、当該解析の結果に基づいてタグが省略されているときには、当該省略されたタグを補充して文書データを処理することを特徴としている。これによりタグが省略された文書データに対しても正常な処理を行うことができる。

【0018】上記従来例の問題点を解決するための請求項8記載の発明は、文書データ処理装置であって、第1のルールで記述された部分文書データと第2のルールに従って記述された部分文書データの少なくとも一方を含む文書データを処理する文書データ処理装置であって、所定の部分文書データを前記第1のルールに従って処理し、出力する第1パーザ手段と、処理対象となった文書データを部分文書データごとに順次解析し、解析の結果に応じて前記第1パーザ手段を起動する動作と、当該解析された部分文書データを第2のルールに従って処理する動作とのいずれかを選択的に行う第2パーザ手段と、を含むことを特徴としている。

【0019】

【発明の実施の形態】本発明の実施の形態について図面を参照しながら説明する。本発明の実施の形態に係る文書データ処理装置は、パーソナルコンピュータであり、具体的には図1に示すように、CPU11と、ブートROM12と、RAM13と、ハードディスク14と、ディスプレイ15と、操作部16と、LANインタフェース17と、外部記憶装置18とから基本的に構成されている。

【0020】CPU11は、電源投入直後にブートROM12に格納されているプログラムをRAM13上にロードして実行する初期化処理を行う。この初期化処理により、CPU11は、ハードディスク14に格納されたオペレーティングシステムをRAM13上にロードし、処理を開始する。そして、CPU11は、操作部16から入力される指示により、文書データの処理を行うブラウザをハードディスク14からRAM13上にロードして処理を行う。このブラウザの処理については、後に詳しく説明する。

【0021】ブートROM12は、CPU11の初期化処理に関連するプログラムを格納している。RAM13は、CPU11のワークメモリとして動作する。ハードディスク14は、CPU11が処理する各種プログラムを格納している。またこのハードディスク14は、CPU11の処理に必要なデータ（例えば事前に設定されたデフォルトDTD等）を格納している。

【0022】ディスプレイ15は、CPU11から入力される指示により、種々のデータを表示出力する。操作部16は、キーボードやマウス等であり、ユーザが行う操作の内容をCPU11に伝達する。LANインタフェース17は、LAN（Local Area Network）又はインターネットを経由してWebサーバに接続されており、CPU11から入力される指示によりネットワークを介してデータを送信し、また、ネットワークを介して到来するデータを受信してCPU11に出力する。

【0023】外部記憶装置18は、フロッピー（登録商標）ディスクや光磁気ディスク等、光学的又は電磁氣的にデータを保持し、コンピュータにより読み取り可能な記録媒体等からデータを読み出してCPU11に出力する。CPU11は、この外部記憶装置18から読み出したデータをハードディスク14に処理プログラムとしてインストールする。

【0024】ここでCPU11の文書データ処理について説明する。本実施の形態に係るCPU11が処理する文書データ処理のためのブラウザプログラムは、図2に示すように、TCP/IPプロトコル解析部21と、XMLパーズ部22と、HTMLパーズ部23と、ブラウザコア部24と、描画部25とから構成されている。また、HTMLパーズ部23は、プレフォーマット処理部31と、CDATA処理部32と、開始タグ処理部33と、終了タグ処理部34とから構成されている。ここ

で、XMLパーズ部22が、本発明のXMLパーザ手段又は第2パーザ手段に、HTMLパーズ部23が本発明のHTMLパーザ手段又は第1パーザ手段にそれぞれ相当している。また、プレフォーマット処理部31とCDATA処理部32とが本発明の特殊タグに対応するパーザ手段に相当し、プレフォーマット処理すべき文書データは、特殊タグ「<PRE>」に関連づけられ、CDATAに関連する特殊タグは、「<SCRIPT>」等である。

【0025】TCP/IPプロトコル処理部21は、TCP/IPプロトコルによってネットワークを経由して文書データを取得し、XMLパーズ部22に出力する。ここで、TCP/IPプロトコル処理部21が取得する文書データは、具体的に図3で示すようなものである。この図3において、文字「<」と「>」とで囲まれている部分（例えば先頭の<html>）がタグと呼ばれる。また、このタグのうち、「</>」で始まるものが終了タグであり、そうでないものが開始タグである。図3に示すように、本実施の形態の文書データ処理装置において想定している文書データは、開始タグとコンテンツデータと終了タグとからなる基本構造が入れ子になっているものである。すなわち、開始タグ<html>と、終了タグ</html>の間のコンテンツには、さらに開始タグ<body>と、終了タグ</body>とに囲まれたコンテンツがあり、さらに、このコンテンツ内にも基本構造が複数含まれている。尚、この図3の文書データにおいては、数多く存在するHTML文書と同様に、例えば「
」タグに対応する終了タグが省略されている。

【0026】また、図3には現れていないが、XMLにおいては、「</>」で終了するタグは、便宜的に終了タグとして扱われるのが一般的である。

【0027】ここで、CPU11がTCP/IPプロトコル処理部21で取得した文書データに対して行うXMLパーズ部22としての処理を図4を参照して説明する。尚、以下の説明において、ハードディスク14には、動作パラメータと、デフォルトデータ型定義に相当するデフォルトDTDとが事前に設定され、格納されているものとする。ここで、動作パラメータとは、図5に示すように、開始タグ処理部33へのポインタ(A)と、終了タグ処理部34へのポインタ(B)と、CDATA処理部32へのポインタ(C)と、CDATAとして処理すべきタグ名の配列(D)と、プレフォーマット処理部31へのポインタ(E)と、プレフォーマット処理部31で処理すべきタグ名の配列(F)と、終了タグの省略の可否を表すフラグ(G)とを関連づけたものであり、デフォルトDTDは、要素宣言と、属性宣言とを含み、要素宣言は、図6(a)に示すように、識別子(H)と、タグ名(I)と、タイプ(J)と、開始タグ及び終了タグの省略可否を表すフラグ(K)と、コンテンツに含まれる可能性のあるタグのリスト(L)とを関連づけたものである。また、属性宣言は、図6(b)に

示すように、識別子(M)と、タグ名(N)と、属性名(O)と、属性値のタイプ(P)とを関連づけたものであり、属性値タイプが列挙型(enumeration)である場合には、さらに取りうる値の配列(Q)が関連づけられている。これらの図5及び図6において配列やリストは、通常広く知られるように、NULLで配列の終了を識別することとしている。

【0028】CPU11は、XMLパーズ部22の処理として、図4に示すように、まず、動作パラメータをハードディスク14からロードし(S1)、デフォルトDTDをハードディスク14からロードする(S2)。そして、文書データを読み込み(S3)、文書データが終了したか否かを調べ(S4)、終了していれば(Yesならば)、処理を終了する。

【0029】また、処理S4において、終了していなければ(Noならば)、読み込んだ文書データがXMLタグであるか否かを調べる(S5)。ここでXMLタグとは、「<」や「>」で開始する特別なタグである。このようなタグとして例えば、「<!-->」で始まるコメントなどがある。この処理S5において、XMLタグであれば(Yesならば)、XMLタグの処理を実行して(S6)、処理S3に戻って処理を続ける(A)。一方、XMLタグでなければ(Noならば)、さらに開始タグであるか否かを調べ(S7)、開始タグであれば(Yesならば)、当該開始タグのタグ名をキーとしてデフォルトDTDを参照し、開始タグ処理部33へのポインタを取得し、読み込んだ文書データを引数として開始タグ処理部33を起動する(S8)。この開始タグ処理部33の動作については後述する。

【0030】そして、開始タグ処理部33の処理が完了すると、CPU11は、XMLパーザの処理を再開し、処理S8で処理した開始タグがCDATAとして処理すべきタグであるか否かを動作パラメータを参照して検査し(S9)、CDATAとして処理すべきタグであれば(Yesであれば)、CDATA処理部32を起動する(S10)。CPU11は、このCDATA処理部32の動作として、対応する終了タグが読み込まれるまでの間、CDATAとして処理し、木構造に追加する処理を行う。そして、CDATA処理部32の動作が完了すると、CPU11は、処理S3に戻ってXMLパーザの処理を続ける。

【0031】一方、処理S9において、CDATAとして処理すべきタグでなければ(Noであれば)、さらに当該開始タグがプレフォーマット処理すべきタグであるか否かを動作パラメータを参照して検査し(S11)、プレフォーマット処理すべきタグであれば(Yesならば)、プレフォーマット処理部31を起動する(S12)。そしてCPU11は、プレフォーマット処理部31の処理として、当該タグに対応する終了タグが読み込まれる間、PCDATAとして木構造に追加する処理を

行う。このプレフォーマット処理部31では、改行コードを検出すると、改行タグ
に置き換える処理を行う。そして、CPU11は、プレフォーマット処理部31の処理が完了すると、処理S3に戻ってXMLパーザの処理を続ける。また、処理S11において、プレフォーマット処理すべきタグでなければ、そのまま処理S3に戻ってXMLパーザの処理を続ける。

【0032】さらに、CPU11は、処理S7において、開始タグでなければ（Noならば）、読み込んだ文書データが終了タグであるか否かを調べる（S13）。そして、終了タグであれば（Yesならば）、終了タグ処理部34を起動し（S14）、終了タグ処理を行い、処理S3に戻って処理を続ける。ここで、終了タグ処理部34の処理内容については、後述する。

【0033】また処理S13において、終了タグでなければ（Noならば）、通常のコンテンツとして処理を行い（S15）、処理S3に戻って処理を続ける。

【0034】ここで、CPU11が行う開始タグ処理部33の動作について説明する。CPU11は、開始タグ処理部33を読み込まれた文書データを引数として起動し、まず、読み込まれたタグの要素を木構造の現時点でポイントしている要素の1レベル下位に追加し、追加した要素を新たに現在の要素としてポイントする。また最初のタグであれば（木構造がなければ）、この最初のタグを木構造の最初のレベル（ルート）として登録し、当該ルートを現在の要素としてポイントする。

【0035】具体的に図4に示した文書データに対し、CPU11は、図7に示す木構造を形成する。すなわち、図4の文書データの1行目の<HTML>を開始タグとして認識し、開始タグ処理部33の処理としてこの「HTML」を木構造のルートとして設定してポイントし（X）、次に2行目の<HEAD>をさらに開始タグとして認識して現在ポイントしている要素（HTML）より1レベル下位に「HEAD」の要素を追加して（Y）、この「HEAD」をポイントする。さらに3行目の「TITLE」も開始タグであるので、さらに1レベル下位に「TITLE」の要素を付加し、引き続き「HOME PAGE」をPCDATAとして付加する（Z）。

【0036】また、ここで、CPU11が行う終了タグ処理部34の動作について説明する。CPU11は、終了タグ処理部34を読み込まれた文書データと木構造中で現在ポイントしている要素とを引数として起動し、XMLの木構造の要素をHTMLとして解析する処理を行って、当該要素をHTML文書に変換してRAM13に格納し、木構造中でポイントしている位置を1レベル上位に設定して、新たに現在の要素としてポイントする。具体的に図4及び図7においては、「TITLE」要素に付加されているPCDATA「HOME PAGE」（Z）の後の「<TITLE>」を終了タグとして認識し、開始タグ「<TITLE>」から終了タグ「</TITLE>」までの基本構造「<TITL

E>HOME PAGE</TITLE>」をHTML文書として解析し、HTML文書としてRAM13に格納する。そして、「TITLE」より1レベル上位の「HEAD」（Y）をポイントするようにして、終了タグの処理を完了する。このようにして、終了タグ処理部34の処理の結果として得られるHTML文書は、図8に示すように、行構造に変換されたものとなる。

【0037】さらにCPU11は、XMLパーズ部22の処理が完了すると、ブラウザコア部24を起動して、RAM13に格納されたHTML文書（HTML及びXMLの混在した文書からXMLパーズ部22及びHTMLパーズ部23の動作によりHTMLに変換された文書）を参照し、このHTML文書に基づいて描画部25に描画の指示を出力する。描画部25は、文書の各行の絶対座標やサイズ、テキストや画像部分、絶対座標やサイズ、テキストに対するフォント、画像を参照する情報としてのURL（Uniform Resource Locators）等を考慮して、ディスプレイ15に描画結果としての文書データの内容を表示する。

【0038】尚、ここまでの説明においては、デフォルトDTDを予めロードしているが、XMLタグ処理において、データ型を宣言するタグである「<!DOCTYPE>」が検出されたときに、当該データ型が特定のもの（例えばHTML3.2）であったときにのみデフォルトDTDをロードするようにしてもよい。また、開始タグ処理部33の動作として、「<HTML>」タグを検出した場合に、それまでにDTD宣言（「<!DOCTYPE>」）を検出していないときにのみ、デフォルトDTDをロードすることとしてもよい。

【0039】さらに、読み込まれた文書データに関連して、当該文書データ内部又は当該文書データから参照されるDTDがある場合には、CPU11は、このDTDをロードして、デフォルトDTDの代わりに、又はデフォルトDTDとともにXMLパーズ部22の処理において利用することも好適である。

【0040】また、動作パラメータにおいて、終了タグの省略の可否を表すフラグにより、特定のタグの省略が可能と設定された場合には、CPU11は、当該開始タグを読み込むと、木構造に追加された要素の1つ上位の要素を取得しておき、ロードされたDTDを参照して、省略可能であるときには、タイプがEMPTYである（「</>」で終了するタグである）か、又はEMPTYでないが要素宣言中コンテンツに含まれる可能性のあるタグのリストにないタグであるときに、終了タグが省略されたものとして終了タグ処理部34を起動する。

【0041】また、省略可能でないタグであるときや、EMPTYでなく、かつ要素宣言中コンテンツに含まれる可能性のあるタグのリストにある場合には、当該タグを木構造に追加して開始タグ処理部33を起動する。

【0042】

12

*【図4】 CPU11のXMLパーサとしての処理を表すフローチャート図である。

【図5】 動作パラメータの一例を表す説明図である。

【図6】 DTDの一例を表す説明図である。

【図 7】 本構造の一例を表す説明図である。

【図8】 行構造の一例を表す説明図である。

【符号の説明】

11 CPU、12 ブートROM、13 RAM、14 ハードディスク、15 ディスプレイ、16 操作

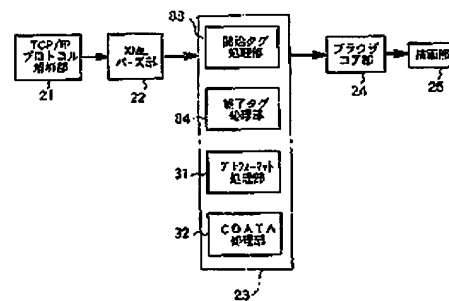
10 部, 17 LANインタフェース、18 外部記憶装置

置. 21 TCP/IPプロトコル解析部、22 XM
Lパース部、23 HTMLパース部、24 ブラウザコ

ア部 25 作画部、31 プレフォーマット処理部、
32 C D A T A 処理部、33 開始タグ処理部、34

終了タグ処理部。

【圖2】



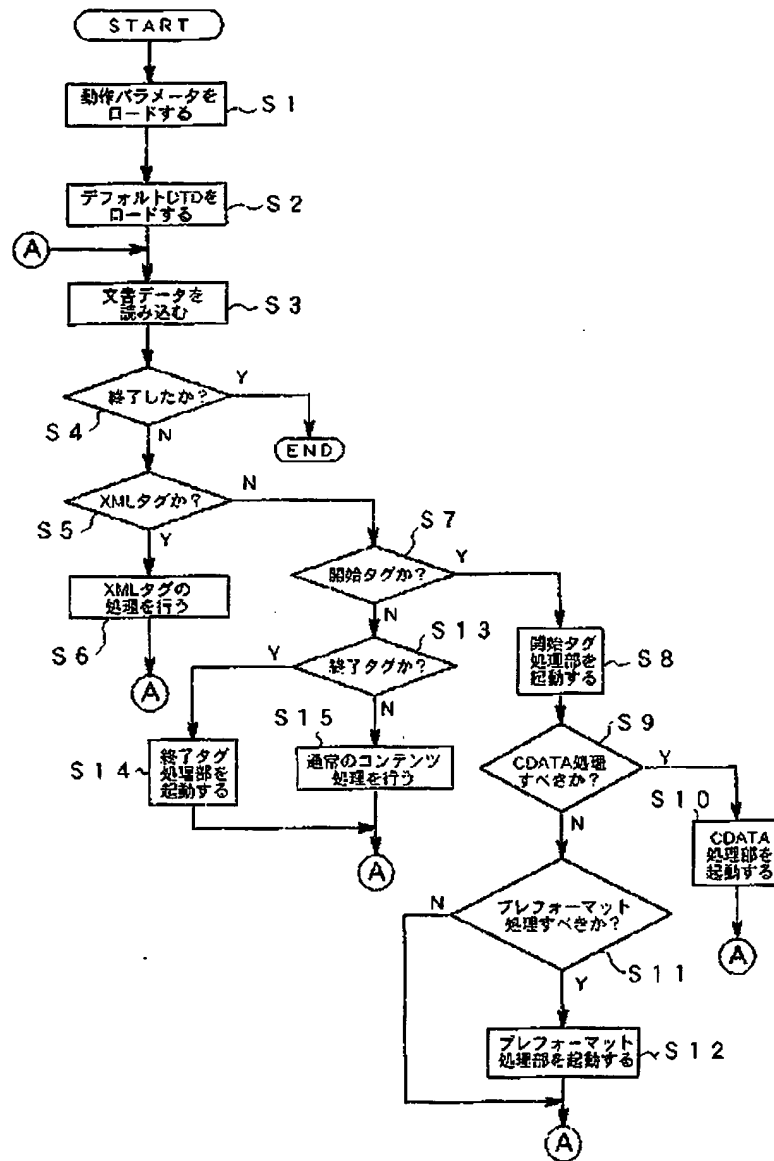
【圖3】

```

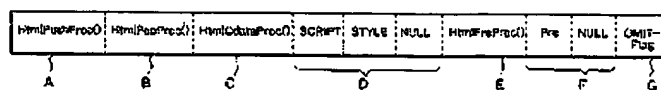
<html>
<head>
<title> HOME PAGE </title>
<style>
H1 {font: bold 20pt serif}
P {font: 12pt Times}
</style>
</head>
<body>
<center>
<h1> Welcome to x x x x x x x x </h1>
<br>
<p>
{A href="http://xxxxxx.xx.jp/img SRC=" enter.gif" } </A>
<br>
{A href="http://xxxxxx.xx.jp/ja" } Click here to enter. </A>
<br>
</center>
</body>
</html>

```

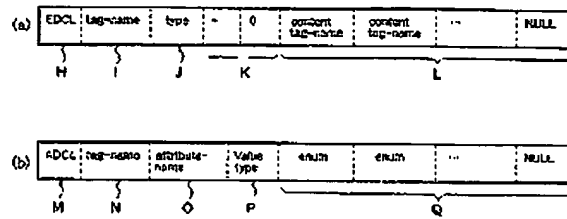
【図4】



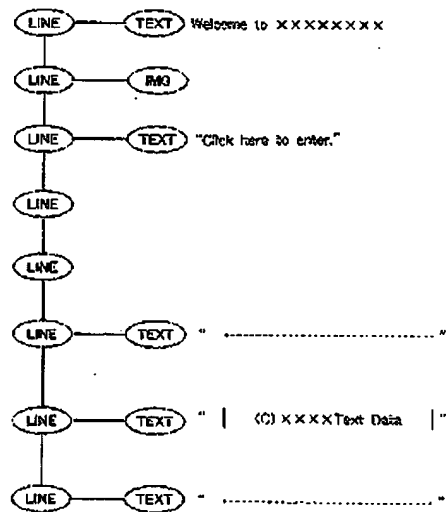
【図5】



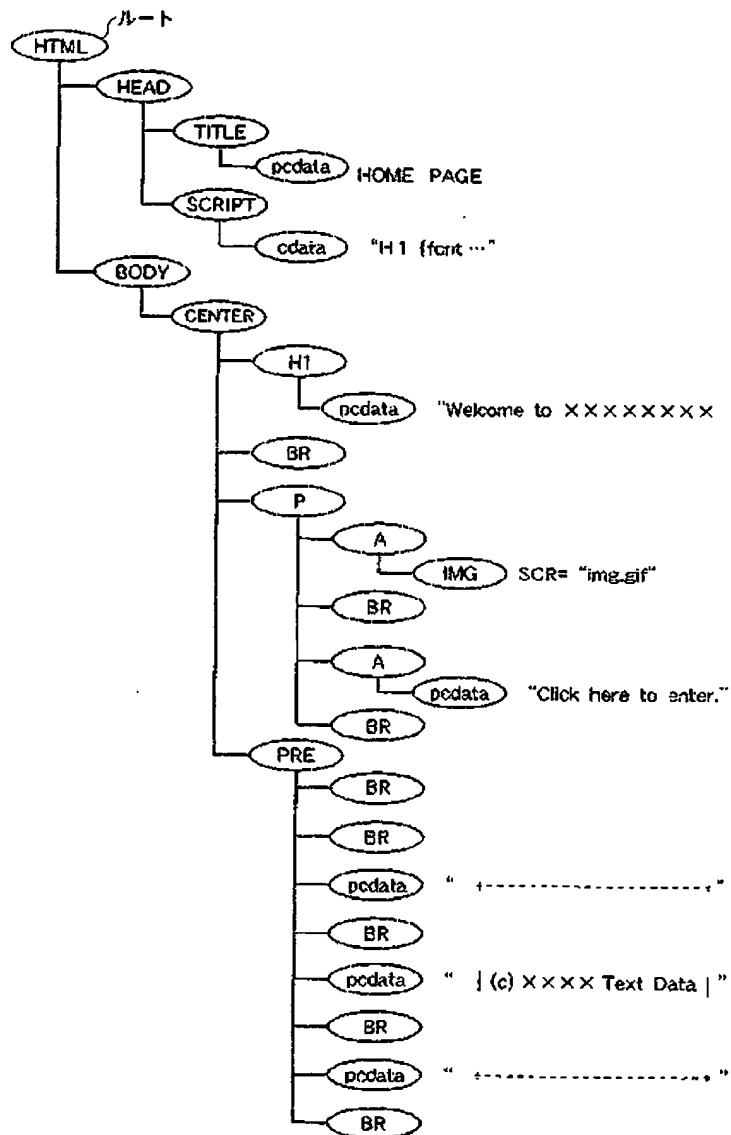
【図6】



【図8】



【図7】



**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

☐ BLACK BORDERS

☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES

☒ FADED TEXT OR DRAWING

☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING

☐ SKEWED/SLANTED IMAGES

☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS

☐ GRAY SCALE DOCUMENTS

☐ LINES OR MARKS ON ORIGINAL DOCUMENT

☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY

☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.